

LAB 20: CONSULTAS EN SQL

Consulta de una tabla completa

Álgebra relacional

materiales

SQL

SELECT *

FROM materiales

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The 'clase_db' database is selected, and the 'materiales' table is highlighted. The main panel shows the table's structure and data. The SQL query 'SELECT * FROM materiales;' is entered in the query box. The results show 24 rows of material data.

clave	descripcion	precio	impuesto
1000	Vanilla 3/16	100	10
1010	Vanilla 4/32	115	11.5
1020	Vanilla 3/17	130	13
1030	Vanilla 4/33	145	14.5
1040	Vanilla 3/18	160	16
1050	Vanilla 4/34	175	17.5
1060	Vanilla 3/19	190	19
1070	Vanilla 4/35	205	20.5
1080	Ladrillos rojos	50	5
1090	Ladrillos grises	35	3.5
1100	Block	30	3
1110	Megablock	40	4
1120	Sillar rosa	100	10
1130	Sillar gris	110	11
1140	Cantera blanca	200	20

Selección

Álgebra relacional

$SL\{clave=1000\}(materiales)$

SQL

SELECT *

FROM materiales

WHERE clave=1000

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The main area displays the results of a SQL query. The query is: `SELECT * FROM materiales WHERE clave=1000;`. The result shows one row with the following data:

clave	descripcion	precio	impuesto
1000	Varilla 3/16	100	10

. The interface includes various navigation and action buttons like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'. The status bar at the bottom indicates 'Showing rows 0 - 0 (1 total, Query took 0.0003 seconds)'.

Proyección

Álgebra relacional

PR{clave,rfc,fecha}(entregan)

SQL

SELECT clave, rfc, fecha

FROM entregan

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a database tree. The 'entregan' table is selected under the 'clase_db' database. The main panel displays a SQL query: `SELECT clave, rfc, fecha FROM entregan;`. Below the query, a status bar indicates 'Showing rows 0 - 24 (87 total, Query took 0.0011 seconds.)'. A table of results is shown with columns 'clave', 'rfc', and 'fecha'. Each row has interactive icons for Edit, Copy, and Delete. The bottom of the interface shows a 'Console' tab.

	clave	rfc	fecha
<input type="checkbox"/> Edit Copy Delete	1000	AAAA800101	2001-12-13
<input type="checkbox"/> Edit Copy Delete	1200	EEEE800101	2003-03-15
<input type="checkbox"/> Edit Copy Delete	1400	AAAA800101	1999-04-07
<input type="checkbox"/> Edit Copy Delete	1010	BBBB800101	1998-07-28
<input type="checkbox"/> Edit Copy Delete	1210	FFFF800101	2000-05-21
<input type="checkbox"/> Edit Copy Delete	1410	BBBB800101	2000-05-18
<input type="checkbox"/> Edit Copy Delete	1020	CCCC800101	2003-12-16
<input type="checkbox"/> Edit Copy Delete	1220	GGGG800101	2005-07-03
<input type="checkbox"/> Edit Copy Delete	1420	CCCC800101	2001-09-09
<input type="checkbox"/> Edit Copy Delete	1030	DDDD800101	1998-01-12
<input type="checkbox"/> Edit Copy Delete	1230	HHHH800101	1998-09-12
<input type="checkbox"/> Edit Copy Delete	1430	DDDD800101	2005-04-30
<input type="checkbox"/> Edit Copy Delete	1040	EEEE800101	1999-12-18
<input type="checkbox"/> Edit Copy Delete	1150	HHHH800101	2002-11-14
<input type="checkbox"/> Edit Copy Delete	1240	AAAA800101	0000-00-00

Reunión Natural

Álgebra relacional

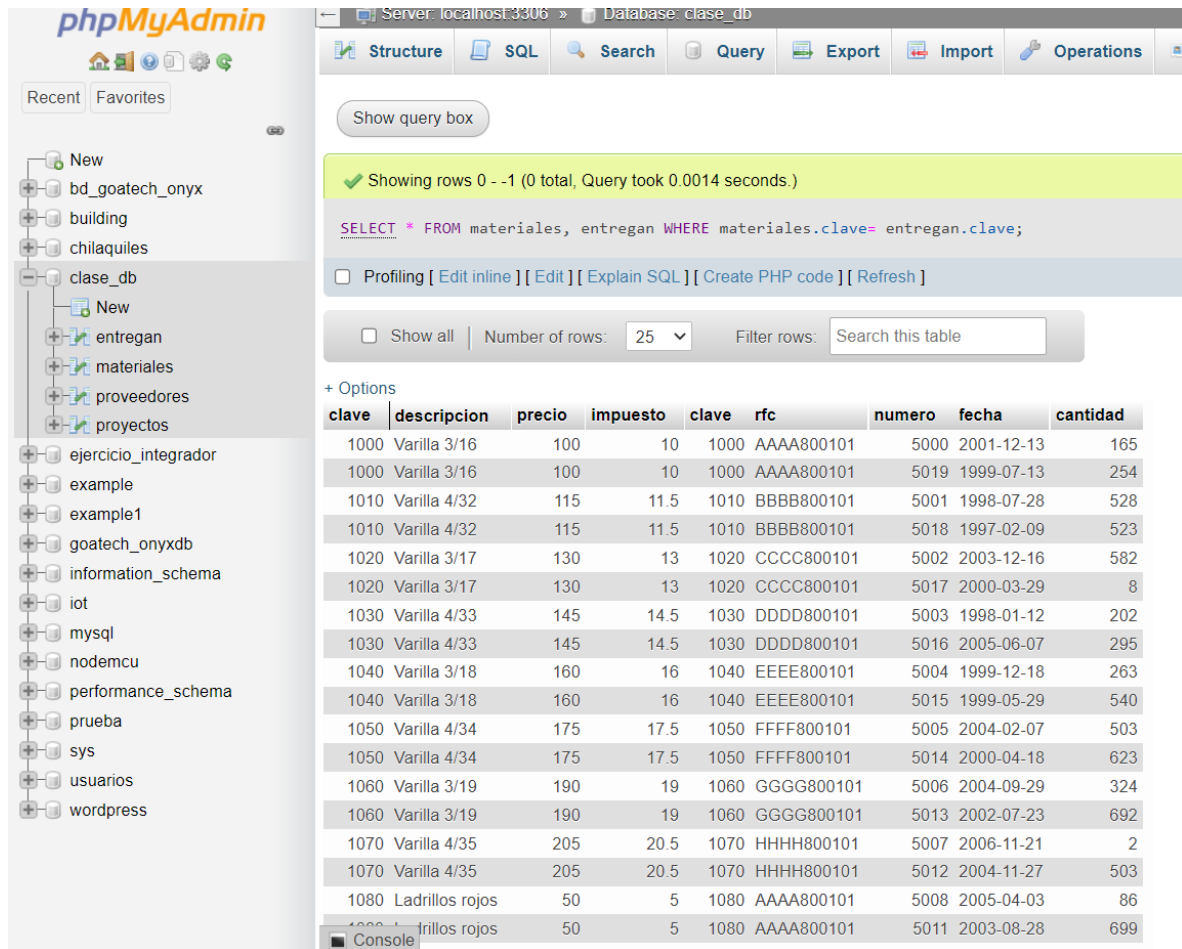
Entregan JN materiales

SQL

SELECT *

FROM materiales, entregan

WHERE materiales.clave= entregan.clave



The screenshot shows the phpMyAdmin interface with the 'clase_db' database selected. A SQL query is executed, displaying a result set of 11 rows. The query is: `SELECT * FROM materiales, entregan WHERE materiales.clave= entregan.clave;`

clave	descripcion	precio	impuesto	clave	rfc	numero	fecha	cantidad
1000	Varilla 3/16	100	10	1000	AAAA800101	5000	2001-12-13	165
1000	Varilla 3/16	100	10	1000	AAAA800101	5019	1999-07-13	254
1010	Varilla 4/32	115	11.5	1010	BBBB800101	5001	1998-07-28	528
1010	Varilla 4/32	115	11.5	1010	BBBB800101	5018	1997-02-09	523
1020	Varilla 3/17	130	13	1020	CCCC800101	5002	2003-12-16	582
1020	Varilla 3/17	130	13	1020	CCCC800101	5017	2000-03-29	8
1030	Varilla 4/33	145	14.5	1030	DDDD800101	5003	1998-01-12	202
1030	Varilla 4/33	145	14.5	1030	DDDD800101	5016	2005-06-07	295
1040	Varilla 3/18	160	16	1040	EEEE800101	5004	1999-12-18	263
1040	Varilla 3/18	160	16	1040	EEEE800101	5015	1999-05-29	540
1050	Varilla 4/34	175	17.5	1050	FFFF800101	5005	2004-02-07	503
1050	Varilla 4/34	175	17.5	1050	FFFF800101	5014	2000-04-18	623
1060	Varilla 3/19	190	19	1060	GGGG800101	5006	2004-09-29	324
1060	Varilla 3/19	190	19	1060	GGGG800101	5013	2002-07-23	692
1070	Varilla 4/35	205	20.5	1070	HHHH800101	5007	2006-11-21	2
1070	Varilla 4/35	205	20.5	1070	HHHH800101	5012	2004-11-27	503
1080	Ladrillos rojos	50	5	1080	AAAA800101	5008	2005-04-03	86
1080	Ladrillos rojos	50	5	1080	AAAA800101	5011	2003-08-28	699

Si algún material no ha se ha entregado ¿Aparecería en el resultado de esta consulta? No, porque para hacer eso, tendrías que haber hecho una diferencia entre los materiales que existen menos los materiales entregados, para ver aquellos que no se han entregado.

Reunión con criterio específico

Álgebra relacional

Entregan JN{entregan.numero <= proyectos.numero} proyectos

SQL

SELECT *

FROM entregan, proyectos

WHERE entregan.numero <= proyectos.numero

The screenshot shows the phpMyAdmin interface for a database named 'clase_db'. The left sidebar displays a tree of databases, with 'clase_db' selected. The main panel shows a SQL query: `SELECT * FROM entregan, proyectos WHERE entregan.numero <= proyectos.numero;`. The query has been executed successfully, showing 0 rows. Below the query, there are options to show all rows, a dropdown for the number of rows (set to 25), and a search filter. The results table is displayed below, showing columns: clave, rfc, numero, fecha, cantidad, numero, and denominacion. The table contains 20 rows of data.

clave	rfc	numero	fecha	cantidad	numero	denominacion
1000	AAAA800101	5000	2001-12-13	165	5000	Vamos Mexico
1200	EEEE800101	5000	2003-03-15	177	5000	Vamos Mexico
1400	AAAA800101	5000	1999-04-07	382	5000	Vamos Mexico
1000	AAAA800101	5000	2001-12-13	165	5001	Aztecon
1200	EEEE800101	5000	2003-03-15	177	5001	Aztecon
1400	AAAA800101	5000	1999-04-07	382	5001	Aztecon
1010	BBBB800101	5001	1998-07-28	528	5001	Aztecon
1210	FFFF800101	5001	2000-05-21	43	5001	Aztecon
1410	BBBB800101	5001	2000-05-18	601	5001	Aztecon
1000	AAAA800101	5000	2001-12-13	165	5002	CIT Campeche
1200	EEEE800101	5000	2003-03-15	177	5002	CIT Campeche
1400	AAAA800101	5000	1999-04-07	382	5002	CIT Campeche
1010	BBBB800101	5001	1998-07-28	528	5002	CIT Campeche
1210	FFFF800101	5001	2000-05-21	43	5002	CIT Campeche
1410	BBBB800101	5001	2000-05-18	601	5002	CIT Campeche
1020	CCCC800101	5002	2003-12-16	582	5002	CIT Campeche
1220	GGGG800101	5002	2005-07-03	24	5002	CIT Campeche
1400	CCCC800101	5002	2001-09-09	603	5002	CIT Campeche

Unión (se ilustra junto con selección)

Álgebra relacional

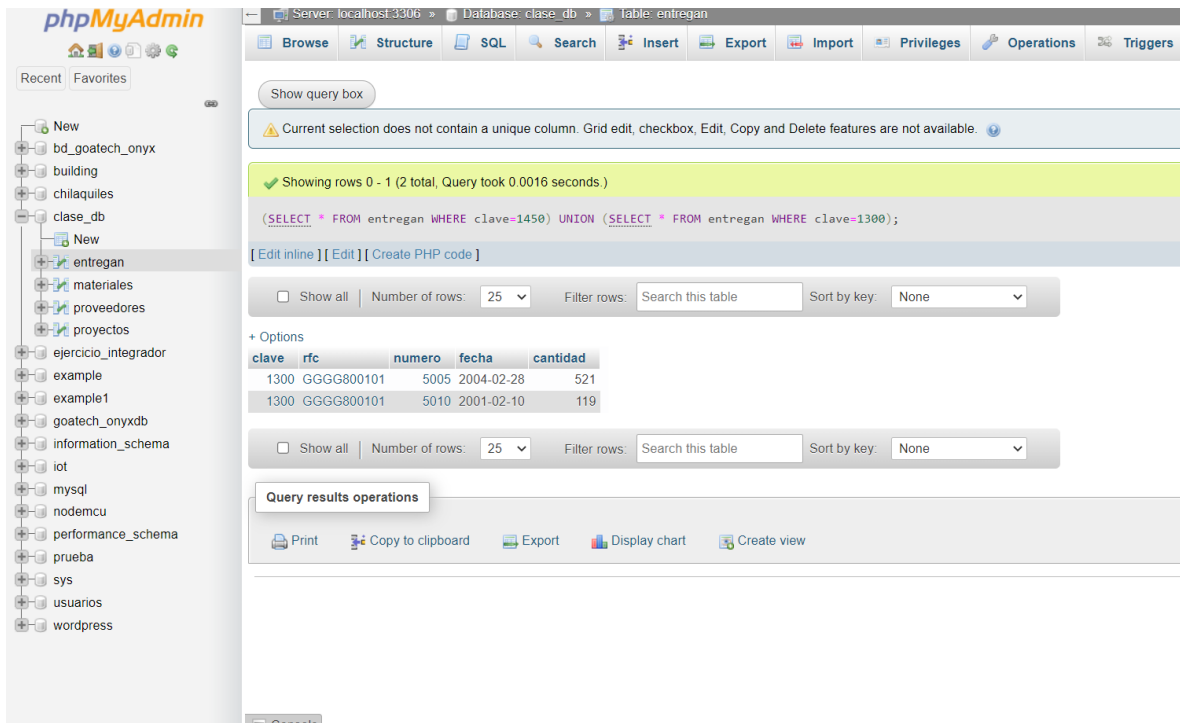
$SL\{clave=1450\}(entregan) \cup SL\{clave=1300\}(entregan)$

SQL

$(SELECT * FROM entregan WHERE clave=1450)$

UNION

$(SELECT * FROM entregan WHERE clave=1300)$



The screenshot shows the phpMyAdmin interface for a database named 'clase_db'. The 'entregan' table is selected. The SQL query entered is: `(SELECT * FROM entregan WHERE clave=1450) UNION (SELECT * FROM entregan WHERE clave=1300);`. The query results show two rows, both with 'clave' 1300, indicating that the first part of the UNION query (where clave=1450) returned no results. The table structure is as follows:

clave	rfc	numero	fecha	cantidad
1300	GGGG800101	5005	2004-02-28	521
1300	GGGG800101	5010	2001-02-10	119

¿Cuál sería una consulta que obtuviera el mismo resultado sin usar el operador Unión? Compruébalo.

`SELECT * FROM entregan WHERE clave=1450 OR clave=1300`

Intersección (se ilustra junto con selección y proyección)

Álgebra relacional

$PR\{clave\}(SL\{numero=5001\}(entregan))$

IN

$PR\{clave\}(SL\{numero=5018\}(entregan))$

SQL

Nota: Debido a que en SQL server no tiene definida alguna palabra reservada que nos permita hacer esto de una manera entendible, veremos esta sección en el siguiente laboratorio con el uso de Subconsultas. Un ejemplo de un DBMS que si tiene la implementación de una palabra reservada para esta función es Oracle, en él si se podría generar la consulta con una sintaxis como la siguiente:

$(SELECT\ clave\ FROM\ entregan\ WHERE\ numero=5001)$

INTERSECT

$(SELECT\ clave\ FROM\ entregan\ WHERE\ numero=5018)$

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a database tree structure. The main panel displays the results of a SQL query. The query is: `(SELECT clave FROM entregan WHERE numero=5001) INTERSECT (SELECT clave FROM entregan WHERE numero=5018)`. The result shows a single row with the value '1010' in the 'clave' column. The interface includes various navigation and action buttons at the top, and a 'Guardar esta consulta en favoritos' (Save this query as favorite) section at the bottom.

phpMyAdmin

Reciente Favoritas

Nueva

- chilaquiles
- clase_db
 - Nueva
 - entregan
 - materiales
 - proveedores
 - proyectos
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test

Servidor: 127.0.0.1 » Base de datos: clase_db » Tabla: entregan

Examinar Estructura SQL Buscar Insertar Exportar Importar Pr

Mostrar ventana de consultas SQL

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0011 segundos.)

`(SELECT clave FROM entregan WHERE numero=5001) INTERSECT (SELECT clave FROM entregan WHERE numero=5018)`

[Editar en línea] [Editar] [Crear código PHP]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

clave

1010

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

Guardar esta consulta en favoritos

Etiqueta: ☐ Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos

Consola

Diferencia

Álgebra relacional

entregan - SL{clave=1000}(entregan)

SQL

(SELECT * FROM entregan)

MINUS

(SELECT * FROM entregan WHERE clave=1000)

Nuevamente, "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.

SELECT * FROM entregan WHERE clave NOT IN (1000)

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The 'entregan' table is selected under the 'clase_db' database. The main panel displays a SQL query: `SELECT * FROM entregan WHERE clave NOT IN (1000);`. Below the query, a status bar indicates 'Showing rows 0 - 24 (85 total, Query took 0.0040 seconds.)'. Below that, there are links for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A pagination bar shows '1' selected, 'Show all' checkbox, and 'Number of rows: 25'. Below the pagination bar is a '+ Options' section. The main area displays a table with 4 columns: 'clave', 'rfc', 'numero', and 'fecha'. Each row has a checkbox, an 'Edit' icon, a 'Copy' icon, and a 'Delete' icon. The table contains 25 rows of data, with the last row being the one with 'clave' 1000, which is the result of the query.

	clave	rfc	numero	fecha
<input type="checkbox"/>	1010	BBBB800101	5001	1998-07
<input type="checkbox"/>	1010	BBBB800101	5018	1997-02
<input type="checkbox"/>	1020	CCCC800101	5002	2003-12
<input type="checkbox"/>	1020	CCCC800101	5017	2000-03
<input type="checkbox"/>	1030	DDDD800101	5003	1998-01
<input type="checkbox"/>	1030	DDDD800101	5016	2005-06
<input type="checkbox"/>	1040	EEEE800101	5004	1999-12
<input type="checkbox"/>	1040	EEEE800101	5015	1999-05
<input type="checkbox"/>	1050	FFFF800101	5005	2004-02
<input type="checkbox"/>	1050	FFFF800101	5014	2000-04
<input type="checkbox"/>	1060	GGGG800101	5006	2004-09
<input type="checkbox"/>	1060	GGGG800101	5013	2002-07
<input type="checkbox"/>	1070	HHHH800101	5007	2006-11
<input type="checkbox"/>	1070	HHHH800101	5012	2004-11
<input type="checkbox"/>	1000	AAAA800101	5008	2005-04

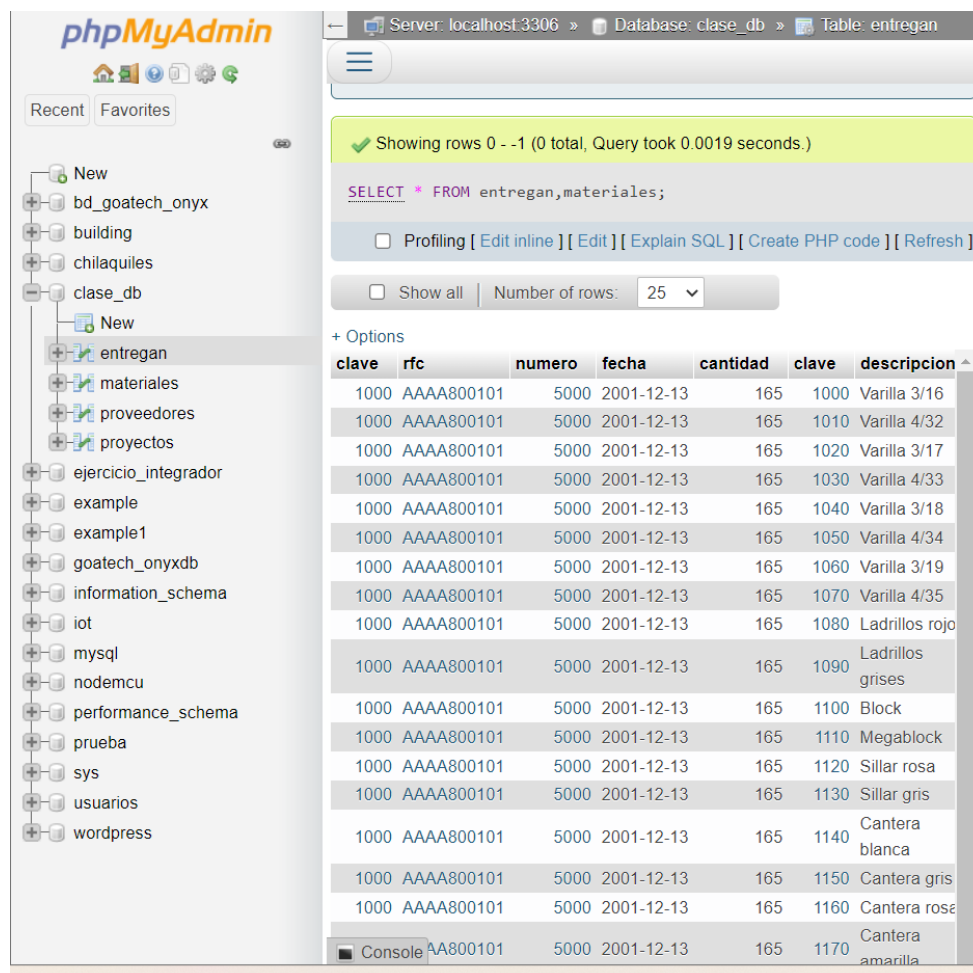
Producto cartesiano

Álgebra relacional

entregan X materiales

SQL

SELECT * FROM entregan,materiales



The screenshot shows the phpMyAdmin interface. The left sidebar displays a database structure with 'clase_db' selected, containing tables like 'entregan' and 'materiales'. The main panel shows the execution of the SQL query 'SELECT * FROM entregan,materiales;'. A message indicates 'Showing rows 0 - -1 (0 total, Query took 0.0019 seconds.)'. Below the query, there are options for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A 'Show all' checkbox and a 'Number of rows' dropdown set to '25' are also visible. The query result is displayed as a table with 7 columns: 'clave', 'rfc', 'numero', 'fecha', 'cantidad', 'clave', and 'descripcion'. The table contains 17 rows of data, representing the Cartesian product of the 'entregan' and 'materiales' tables. The first column 'clave' is constant at 1000, 'rfc' is constant at AAAA800101, 'numero' is constant at 5000, and 'fecha' is constant at 2001-12-13. The 'cantidad' column is constant at 165. The last two columns, 'clave' and 'descripcion', list various materials like 'Varilla 3/16', 'Ladrillos rojo', 'Block', etc.

clave	rfc	numero	fecha	cantidad	clave	descripcion
1000	AAAA800101	5000	2001-12-13	165	1000	Varilla 3/16
1000	AAAA800101	5000	2001-12-13	165	1010	Varilla 4/32
1000	AAAA800101	5000	2001-12-13	165	1020	Varilla 3/17
1000	AAAA800101	5000	2001-12-13	165	1030	Varilla 4/33
1000	AAAA800101	5000	2001-12-13	165	1040	Varilla 3/18
1000	AAAA800101	5000	2001-12-13	165	1050	Varilla 4/34
1000	AAAA800101	5000	2001-12-13	165	1060	Varilla 3/19
1000	AAAA800101	5000	2001-12-13	165	1070	Varilla 4/35
1000	AAAA800101	5000	2001-12-13	165	1080	Ladrillos rojo
1000	AAAA800101	5000	2001-12-13	165	1090	Ladrillos grises
1000	AAAA800101	5000	2001-12-13	165	1100	Block
1000	AAAA800101	5000	2001-12-13	165	1110	Megablock
1000	AAAA800101	5000	2001-12-13	165	1120	Sillar rosa
1000	AAAA800101	5000	2001-12-13	165	1130	Sillar gris
1000	AAAA800101	5000	2001-12-13	165	1140	Cantera blanca
1000	AAAA800101	5000	2001-12-13	165	1150	Cantera gris
1000	AAAA800101	5000	2001-12-13	165	1160	Cantera rosa
1000	AAAA800101	5000	2001-12-13	165	1170	Cantera amarilla

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales? El número total de tuplas en producto cartesiano es el # de tuplas de materiales por el # de tuplas de entregan (AxB).

Construcción de consultas a partir de una especificación

Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

Recuerda que la fecha puede indicarse como '01-JAN-2000' o '01/01/00'.

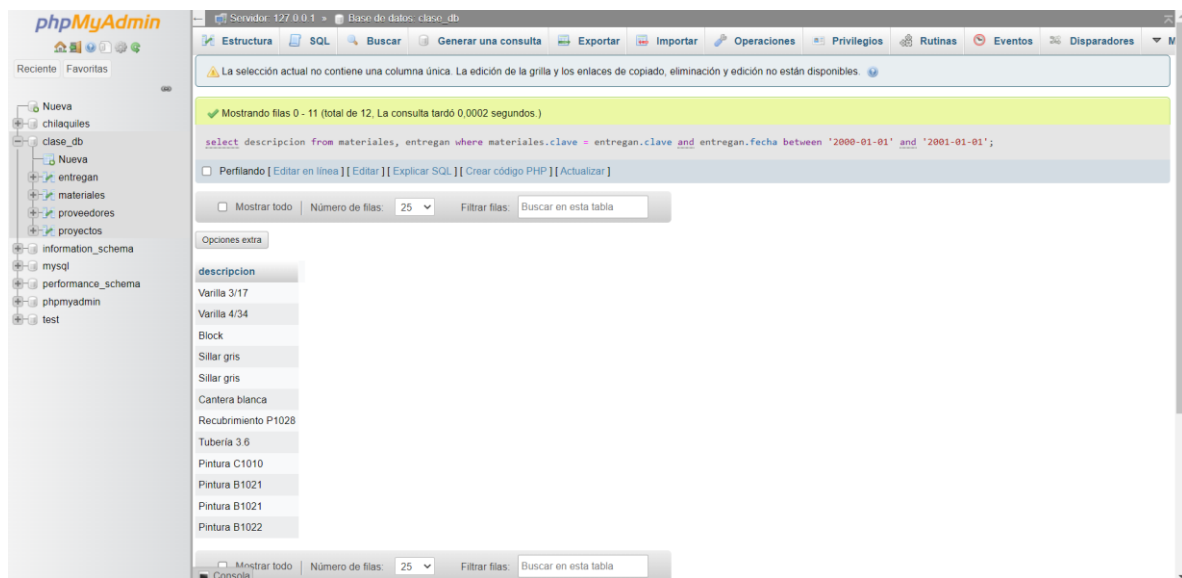
Importante: Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa que vas a trabajar con ese formato de fechas.

SELECT descripcion

FROM materiales, entregan

WHERE materiales.clave = entregan.clave

AND entregan.fecha BETWEEN '2000-01-01' and '2001-01-01';



¿Por qué aparecen varias veces algunas descripciones de material? Porque se entregó un mismo material en varias fechas en el año 2000.

Uso del calificador distinct

En el resultado anterior, observamos que una misma descripción de material aparece varias veces.

Agrega la palabra `distinct` inmediatamente después de la palabra `select` a la consulta que planteaste antes.

`SELECT DISTINCT descripcion`

`FROM materiales, entregan`

`WHERE materiales.clave = entregan.clave`

`AND entregan.fecha BETWEEN '2000-01-01' and '2001-01-01';`

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The main panel displays a SQL query and its results. The query is: `SELECT DISTINCT descripcion FROM materiales, entregan WHERE materiales.clave = entregan.clave`. The results table has one column, 'descripcion', and lists 10 unique material descriptions. The interface includes navigation tabs like 'Estructura', 'SQL', 'Buscar', and 'Generar una consulta'. A status bar at the bottom shows 'Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0002 segundos.)'.

phpMyAdmin

Servidor: 127.0.0.1 » Base de datos: clase_db

Estructura SQL Buscar Generar una consulta Exportar Importar

Mostrar ventana de consultas SQL

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación...

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0002 segundos.)

`SELECT DISTINCT descripcion FROM materiales, entregan WHERE materiales.clave = entregan.clave`

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

descripcion
Varilla 3/17
Varilla 4/34
Block
Sillar gris
Cantera blanca
Recubrimiento P1028
Tubería 3.6
Pintura C1010
Pintura B1021
Pintura B1022

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Consola

¿Qué resultado obtienes en esta ocasión? Obtenemos los materiales cuya descripción no se repite en el año 2000.

Ordenamientos.

Si al final de una sentencia select se agrega la cláusula

order by campo [desc] [,campo [desc] ...]

donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.

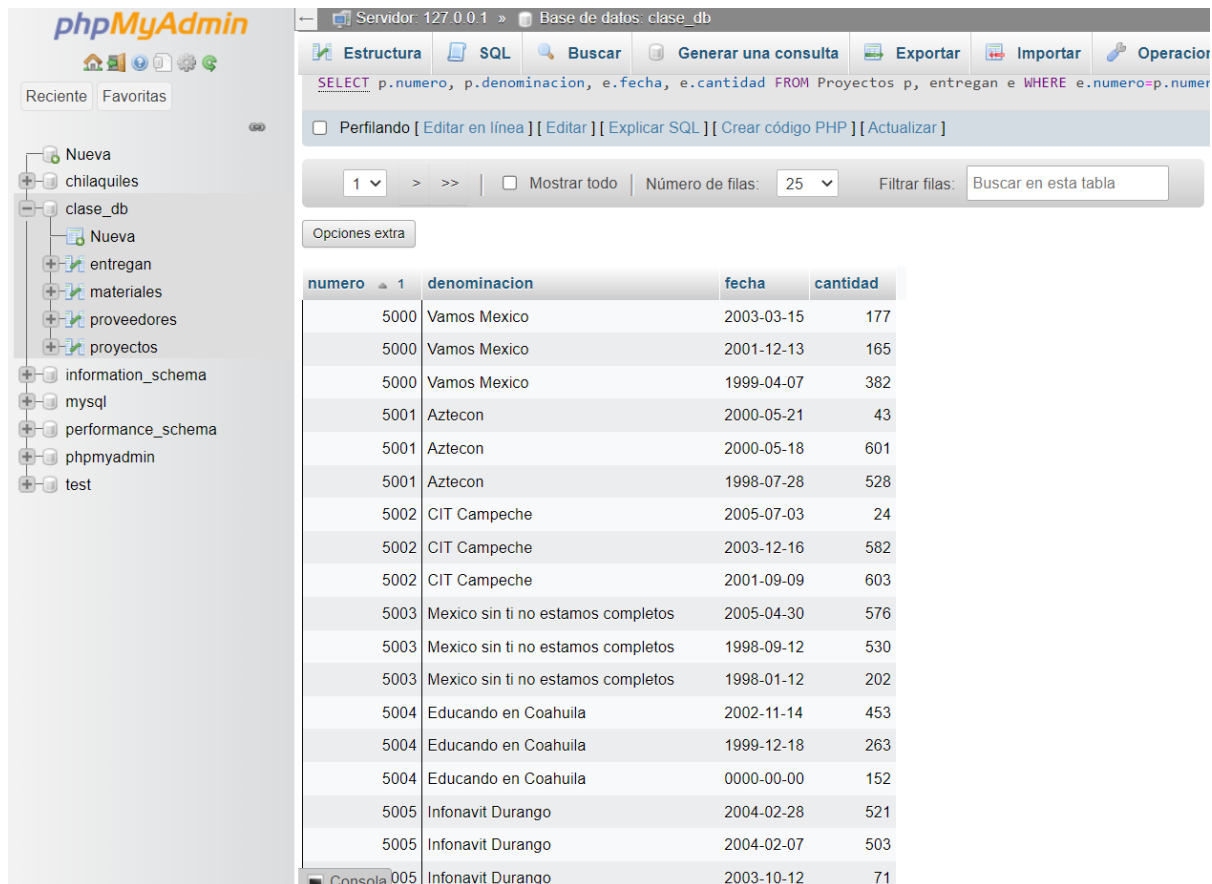
Obtén los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto, presentando las fechas de la más reciente a la más antigua.

SELECT p.numero, p.denominacion, e.fecha, e.cantidad

FROM Proyectos p, entregan e

WHERE e.numero=p.numero

ORDER BY p.numero, e.fecha DESC



The screenshot shows the phpMyAdmin interface. On the left is the database structure tree with 'clase_db' selected. The main panel displays a SQL query: `SELECT p.numero, p.denominacion, e.fecha, e.cantidad FROM Proyectos p, entregan e WHERE e.numero=p.numero`. Below the query, there are options for 'Perfilando' and a table view selector set to '1'. The results are shown in a table with 4 columns: 'numero', 'denominacion', 'fecha', and 'cantidad'. The table contains 20 rows of data, ordered by project number and then by date in descending order.

numero	denominacion	fecha	cantidad
5000	Vamos Mexico	2003-03-15	177
5000	Vamos Mexico	2001-12-13	165
5000	Vamos Mexico	1999-04-07	382
5001	Aztecon	2000-05-21	43
5001	Aztecon	2000-05-18	601
5001	Aztecon	1998-07-28	528
5002	CIT Campeche	2005-07-03	24
5002	CIT Campeche	2003-12-16	582
5002	CIT Campeche	2001-09-09	603
5003	Mexico sin ti no estamos completos	2005-04-30	576
5003	Mexico sin ti no estamos completos	1998-09-12	530
5003	Mexico sin ti no estamos completos	1998-01-12	202
5004	Educando en Coahuila	2002-11-14	453
5004	Educando en Coahuila	1999-12-18	263
5004	Educando en Coahuila	0000-00-00	152
5005	Infonavit Durango	2004-02-28	521
5005	Infonavit Durango	2004-02-07	503
5005	Infonavit Durango	2003-10-12	71

Uso de expresiones.

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo, en una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

+ Suma

- Resta

* Producto

/ División

Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre"). Para SQL Server también pueden utilizarse comillas simples.

Operadores de cadena (LIKE)

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.

```
SELECT * FROM materiales where Descripcion LIKE 'Si%';
```

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The main area displays the results of a SQL query. The query is: `SELECT * FROM materiales where Descripcion LIKE 'Si%';`. The results show two rows of data from the 'materiales' table.

	clave	descripcion	precio	impuesto
<input type="checkbox"/>	1120	Sillar rosa	100	10
<input type="checkbox"/>	1130	Sillar gris	110	11

¿Qué resultado obtienes? Se obtiene la tabla completa de materiales donde la descripción comience con 'Si'.

Explica que hace el símbolo '%'. Hace un patrón que ignora los valores de 0 o más caracteres.

¿Qué sucede si la consulta fuera : LIKE 'Si' ? Da todas las descripciones de los materiales que sean 'Si'.

¿Qué resultado obtienes? Nada, porque no hay ni un solo material cuya descripción sea únicamente 'Si'.

Explica a qué se debe este comportamiento. Esto se debe debido a que el módulo hace una búsqueda por patrones, entonces si en la consulta le pones LIKE 'Patrón' te va a arrojar consultas que sigan el patrón definido, pero si no especificas el patrón y únicamente poner la palabra, va a buscar entradas con esa palabra exacta, pues no es búsqueda con patrones.

Operadores de cadena (CONCATENACIÓN)

Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.

Su sintaxis es : Expresión + Expresión.

Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:

```
SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;
```

```
DECLARE @foo varchar(40);
```

```
DECLARE @bar varchar(40);
```

```
SET @foo = '¿Que resultado';
```

```
SET @bar = ' ¿¿¿???'
```

```
SET @foo += ' obtienes?';
```

```
PRINT @foo + @bar;
```

¿Qué resultado obtienes de ejecutar el siguiente código? No hay ningún código que se pueda correr para hacer consultas, por ende, no se da ningún resultado.

¿Para qué sirve DECLARE? Es para hacer una variable local.

¿Cuál es la función de @foo? Definir una variable varchar(40)

¿Que realiza el operador SET? Sirve para ponerle un valor a una variable.

Operadores de cadena ([], [^] y _).

[] - Busca coincidencia dentro de un intervalo o conjunto dado. Estos caracteres se pueden utilizar para buscar coincidencias de patrones como sucede con LIKE.

[^] - En contra parte, este operador coincide con cualquier caracter que no se encuentre dentro del intervalo o del conjunto especificado.

_ - El operador _ o guion bajo, se utiliza para coincidir con un caracter de una comparación de cadenas.

Ahora explica el comportamiento, función y resultado de cada una de las siguientes consultas:

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';

Muestra las RFC de entregan que comiencen con A,B,C o D (rango)% patrón.

SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';

Muestra las RFC de entregan donde la RFC no comience con A.

SELECT Numero FROM Entregan WHERE Numero LIKE '____6';

Muestra el número de entregan donde se muestren todos los número de 4 cifras que terminen con 6.

Operadores compuestos.

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

Operadores Lógicos.

Los operadores lógicos comprueban la verdad de una condición, al igual que los operadores de comparación, devuelven un tipo de dato booleano (True, false o unknown).

ALL Es un operador que compara un valor numérico con un conjunto de valores representados por un subquery. La condición es verdadera cuando todo el conjunto cumple la condición.

ANY o **SOME** Es un operador que compara un valor numérico con un conjunto de valores. La condición es verdadera cuando al menos un dato del conjunto cumple la condición

La sintaxis para ambos es: valor_numerico {operador de comparación} subquery

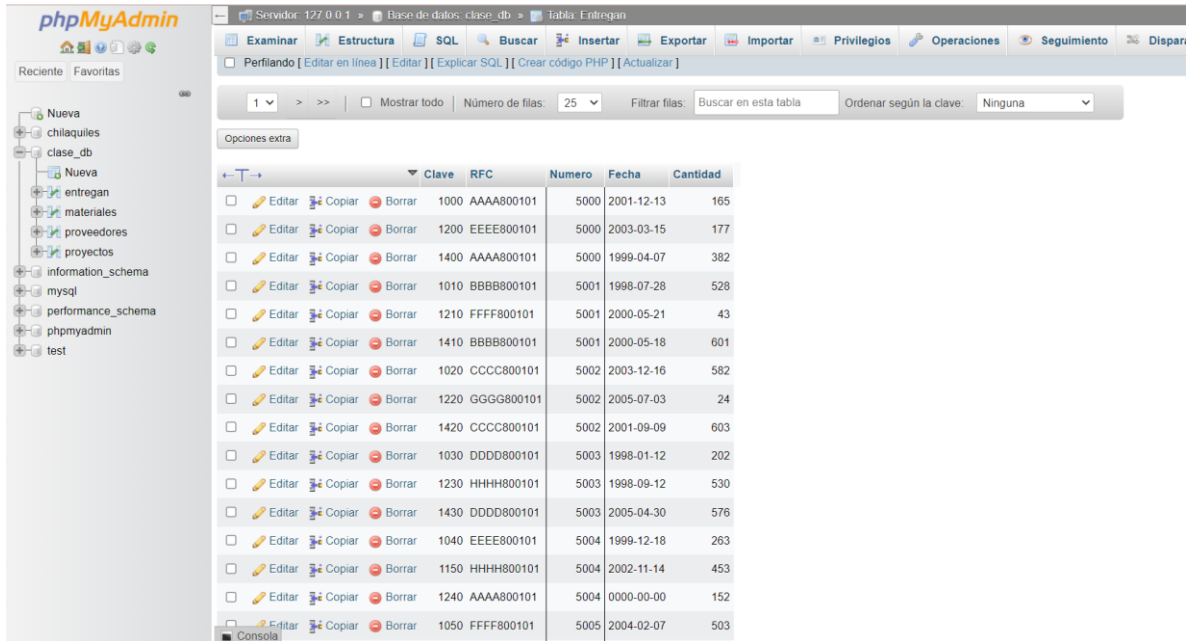
BETWEEN Es un operador para especificar intervalos. Una aplicación muy común de dicho operador son intervalos de fechas.

María Fernanda Moreno Gómez A01708653

SELECT Clave,RFC,Numero,Fecha,Cantidad

FROM Entregan

WHERE Numero Between 5000 and 5010;



	Clave	RFC	Numero	Fecha	Cantidad
<input type="checkbox"/>	1000	AAAA800101	5000	2001-12-13	165
<input type="checkbox"/>	1200	EEEE800101	5000	2003-03-15	177
<input type="checkbox"/>	1400	AAAA800101	5000	1999-04-07	382
<input type="checkbox"/>	1010	BBBB800101	5001	1998-07-28	528
<input type="checkbox"/>	1210	FFFF800101	5001	2000-05-21	43
<input type="checkbox"/>	1410	BBBB800101	5001	2000-05-18	601
<input type="checkbox"/>	1020	CCCC800101	5002	2003-12-16	582
<input type="checkbox"/>	1220	GGGG800101	5002	2005-07-03	24
<input type="checkbox"/>	1420	CCCC800101	5002	2001-09-09	603
<input type="checkbox"/>	1030	DDDD800101	5003	1998-01-12	202
<input type="checkbox"/>	1230	HHHH800101	5003	1998-09-12	530
<input type="checkbox"/>	1430	DDDD800101	5003	2005-04-30	576
<input type="checkbox"/>	1040	EEEE800101	5004	1999-12-18	263
<input type="checkbox"/>	1150	HHHH800101	5004	2002-11-14	453
<input type="checkbox"/>	1240	AAAA800101	5004	0000-00-00	152
<input type="checkbox"/>	1050	FFFF800101	5005	2004-02-07	503

¿Cómo filtrarías rangos de fechas? En la parte de WHERE se pone un BETWEEN entre las fechas deseadas.

EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.

SELECT RFC,Cantidad, Fecha,Numero

FROM Entregan

WHERE Numero Between 5000 and 5010 AND

Exists (SELECT RFC

FROM Proveedores

WHERE RazonSocial LIKE 'La%' and Entregan.RFC = Proveedores.RFC);

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The main area displays a SQL query and its results. The query is: `SELECT RFC,Cantidad, Fecha,Numero FROM Entregan WHERE Numero Between 5000 and 5010 AND Exists (SELECT RFC FROM Proveedores WHERE RazonSoc: Proveedores.RFC);`. The results table has 11 rows and 5 columns: RFC, Cantidad, Fecha, and Numero. Each row includes action links (Editar, Copiar, Borrar) and a 'Consola' button at the bottom.

	RFC	Cantidad	Fecha	Numero
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	165	2001-12-13	5000
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	86	2005-04-03	5008
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	152	0000-00-00	5004
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	382	1999-04-07	5000
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	116	2005-04-21	5010
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	582	2003-12-16	5002
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	466	2000-12-07	5009
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	699	2001-11-19	5010
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	460	2001-04-09	5006
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	631	2001-07-28	5009
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	CCCC800101	603	2001-09-09	5002

¿Qué hace la consulta? Selecciona rfc, cantidad, fecha y número de entregan donde el número está entre 500 y 5010 y que además su razón social empiece con 'La'.

¿Qué función tiene el paréntesis () después de EXISTS? Es para indicar que existe una subconsulta la cual se debe de cumplir también como parte de la solución.

IN Especifica si un valor dado tiene coincidencias con algún valor de una subconsulta. **NOTA:** Se utiliza dentro del WHERE pero debe contener un parametro. Ejemplo: Where proyecto.id IN Lista_de_Proyectos_Subquery

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN

```
SELECT RFC,Cantidad, Fecha,Numero
FROM Entregan
WHERE Numero Between 5000 and 5010 AND
RFC IN (SELECT RFC
FROM Proveedores
WHERE RazonSocial LIKE 'La%');
```

The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'clase_db' selected. The main panel displays the 'Entregan' table. A SQL query is entered in the query box: `SELECT RFC,Cantidad, Fecha,Numero FROM Entregan WHERE Numero Between 5000 and 5010 AND RFC IN (SELECT RFC FROM Proveedores`. Below the query box, the results of the query are shown in a table with 4 columns: RFC, Cantidad, Fecha, and Numero. The table contains 12 rows of data. At the bottom of the interface, there are buttons for 'Seleccionar todo', 'Editar', 'Copiar', 'Borrar', and 'Exportar'.

	RFC	Cantidad	Fecha	Numero
<input type="checkbox"/>	AAAA800101	165	2001-12-13	5000
<input type="checkbox"/>	AAAA800101	86	2005-04-03	5008
<input type="checkbox"/>	AAAA800101	152	0000-00-00	5004
<input type="checkbox"/>	AAAA800101	382	1999-04-07	5000
<input type="checkbox"/>	AAAA800101	116	2005-04-21	5010
<input type="checkbox"/>	CCCC800101	582	2003-12-16	5002
<input type="checkbox"/>	CCCC800101	466	2000-12-07	5009
<input type="checkbox"/>	CCCC800101	699	2001-11-19	5010
<input type="checkbox"/>	CCCC800101	460	2001-04-09	5006
<input type="checkbox"/>	CCCC800101	631	2001-07-28	5009
<input type="checkbox"/>	CCCC800101	603	2001-09-09	5002
<input type="checkbox"/>	CCCC800101	278	1999-05-05	5008

NOT Simplemente niega la entrada de un valor booleano.

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

```
SELECT RFC,Cantidad, Fecha,Numero
FROM Entregan
WHERE Numero Between 5000 and 5010 AND
RFC NOT IN (SELECT RFC
FROM Proveedores
```

WHERE RazonSocial NOT LIKE 'La%' and Entregan.RFC = Proveedores.RFC);

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a database tree. The main area displays a table named 'Entregan' with the following SQL query: `SELECT RFC,Cantidad, Fecha,Numero FROM Entregan WHERE Numero Between 5000 and 5010 AND RFC NOT IN (SELECT RFC FROM Proveedores Entregan.RFC = Proveedores.RFC);`. The results table has 12 rows, showing columns RFC, Cantidad, Fecha, and Numero. The first few rows are: (AAAA800101, 165, 2001-12-13, 5000), (AAAA800101, 382, 1999-04-07, 5000), (CCCC800101, 582, 2003-12-16, 5002), (CCCC800101, 603, 2001-09-09, 5002), (AAAA800101, 152, 0000-00-00, 5004), (CCCC800101, 460, 2001-04-09, 5006), (AAAA800101, 86, 2005-04-03, 5008), (CCCC800101, 278, 1999-05-05, 5008), (CCCC800101, 466, 2000-12-07, 5009), (CCCC800101, 631, 2001-07-28, 5009), (CCCC800101, 699, 2001-11-19, 5010), and (AAAA800101, 116, 2005-04-21, 5010).

	RFC	Cantidad	Fecha	Numero
<input type="checkbox"/>	AAAA800101	165	2001-12-13	5000
<input type="checkbox"/>	AAAA800101	382	1999-04-07	5000
<input type="checkbox"/>	CCCC800101	582	2003-12-16	5002
<input type="checkbox"/>	CCCC800101	603	2001-09-09	5002
<input type="checkbox"/>	AAAA800101	152	0000-00-00	5004
<input type="checkbox"/>	CCCC800101	460	2001-04-09	5006
<input type="checkbox"/>	AAAA800101	86	2005-04-03	5008
<input type="checkbox"/>	CCCC800101	278	1999-05-05	5008
<input type="checkbox"/>	CCCC800101	466	2000-12-07	5009
<input type="checkbox"/>	CCCC800101	631	2001-07-28	5009
<input type="checkbox"/>	CCCC800101	699	2001-11-19	5010
<input type="checkbox"/>	AAAA800101	116	2005-04-21	5010

El Operador **TOP**, es un operador que recorre la entrada, un query, y sólo devuelve el primer número o porcentaje específico de filas basado en un criterio de ordenación si es posible.

¿Qué hace la siguiente sentencia? Explica por qué.

SELECT TOP 2 * FROM Proyectos

Regresaría los primeros dos renglones de la tabla proyectos, pues TOP indica primer, y el dos quiere indicar que los primeros dos.

¿Qué sucede con la siguiente consulta? Explica por qué.

SELECT TOP Numero FROM Proyectos

Número es una columna de proyectos, por lo que no se puede poner esta para indicar el número de renglones ya que no es un número. Regresa error.

Modificando la estructura de un tabla existente.

Agrega a la tabla materiales la columna PorcentajImpuesto con la instrucción:

ALTER TABLE materiales ADD PorcentajImpuesto NUMERIC(6,2);

A fin de que los materiales tengan un impuesto, les asignaremos impuestos ficticios basados en sus claves con la instrucción:

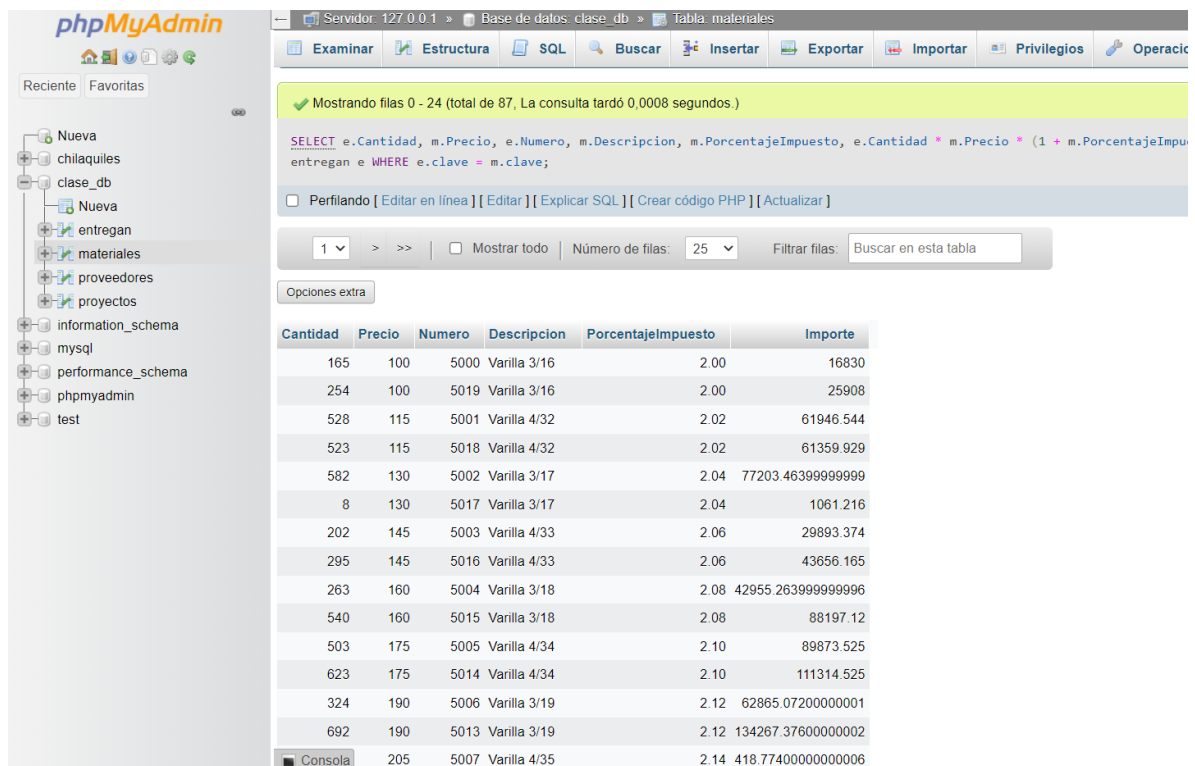
UPDATE materiales SET PorcentajImpuesto = 2*clave/1000;

esto es, a cada material se le asignará un impuesto igual al doble de su clave dividida entre diez.

Revisa la tabla de materiales para que compruebes lo que hicimos anteriormente.

¿Qué consulta usarías para obtener el importe de las entregas es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega y el precio del material y el impuesto asignado?

```
SELECT      e.Cantidad,      m.Precio,      e.Numero,      m.Descripcion,
m.PorcentajImpuesto,
e.Cantidad * m.Precio * (1 + m.PorcentajImpuesto / 100) AS Importe
FROM materiales m, entregan e
WHERE e.clave = m.clave;
```



The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'clase_db' selected. The main panel displays a SQL query and its results. The query is: `SELECT e.Cantidad, m.Precio, e.Numero, m.Descripcion, m.PorcentajeImpuesto, e.Cantidad * m.Precio * (1 + m.PorcentajeImpuesto / 100) AS Importe FROM entregan e WHERE e.clave = m.clave;` The results table has 6 columns: Cantidad, Precio, Numero, Descripcion, PorcentajeImpuesto, and Importe. It contains 24 rows of data.

Cantidad	Precio	Numero	Descripcion	PorcentajeImpuesto	Importe
165	100	5000	Varilla 3/16	2.00	16830
254	100	5019	Varilla 3/16	2.00	25908
528	115	5001	Varilla 4/32	2.02	61946.544
523	115	5018	Varilla 4/32	2.02	61359.929
582	130	5002	Varilla 3/17	2.04	77203.46399999999
8	130	5017	Varilla 3/17	2.04	1061.216
202	145	5003	Varilla 4/33	2.06	29893.374
295	145	5016	Varilla 4/33	2.06	43656.165
263	160	5004	Varilla 3/18	2.08	42955.263999999996
540	160	5015	Varilla 3/18	2.08	88197.12
503	175	5005	Varilla 4/34	2.10	89873.525
623	175	5014	Varilla 4/34	2.10	111314.525
324	190	5006	Varilla 3/19	2.12	62865.07200000001
692	190	5013	Varilla 3/19	2.12	134267.37600000002
205	205	5007	Varilla 4/35	2.14	418.77400000000006

Creación de vistas

La sentencia:

```
Create view nombrevista (nombrecolumna1 , nombrecolumna2 ,...,  
nombrecolumna3 )
```

```
as select...
```

Permite definir una vista. Una vista puede pensarse como una consulta etiquetada con un nombre, ya que en realidad al referirnos a una vista el DBMS realmente ejecuta la consulta asociada a ella, pero por la cerradura del álgebra relacional, una consulta puede ser vista como una nueva relación o tabla, por lo que es perfectamente válido emitir la sentencia

```
select * from nombrevista
```

¡Como si nombrevista fuera una tabla!

Comprueba lo anterior, creando vistas para cinco de las consultas que planteaste anteriormente en la práctica . Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

```
CREATE VIEW Entregados2000
```

```
AS SELECT descripcion
```

```
FROM materiales, entregan
```

```
WHERE materiales.clave = entregan.clave
```

```
AND entregan.fecha BETWEEN '2000-01-01' and '2001-01-01';
```

```
(SELECT * FROM Entregados2000)
```

```
CREATE VIEW ProyectosRZ
```

```
AS SELECT RFC,Cantidad, Fecha,Numero
```

```
FROM Entregan
```

```
WHERE Numero Between 5000 and 5010 AND
```

```
RFC IN (SELECT RFC
```

```
FROM Proveedores
```

```
WHERE RazonSocial LIKE 'La%');
```

```
(SELECT * FROM ProyectosRZ)
```

```
CREATE VIEW MatPreclmp
AS SELECT e.Cantidad, m.Precio, e.Numero, m.Descripcion,
m.PorcentajImpuesto,
e.Cantidad * m.Precio * (1 + m.PorcentajImpuesto / 100) AS Importe
FROM materiales m, entregan e
WHERE e.clave = m.clave;
(SELECT * FROM MatPreclmp)
```

```
CREATE VIEW DescrSi
AS SELECT * FROM materiales where Descripcion LIKE 'Si%';
(SELECT * FROM DescrSi)
```

```
CREATE VIEW EntregasNot1000
AS SELECT * FROM entregan WHERE clave NOT IN (1000);
(SELECT * FROM EntregasNot1000)
```


Reporte

A continuación, se te dan muchos enunciados de los cuales deberás generar su correspondiente consulta.

En el reporte incluye la sentencia, una muestra de la salida (dos o tres renglones) y el número de renglones que SQL Server reporta al final de la consulta.

1. Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos".

```
SELECT m.clave, m.descripcion
FROM materiales m, entregan e, proyectos pr
WHERE m.clave=e.clave
AND pr.numero=e.numero
AND pr.denominacion= 'Mexico sin ti no estamos completos';
```

	clave	descripcion
<input type="checkbox"/>  Editar  Copiar  Borrar	1030	Varilla 4/33
<input type="checkbox"/>  Editar  Copiar  Borrar	1230	Cemento
<input type="checkbox"/>  Editar  Copiar  Borrar	1430	Pintura B1022

Renglones: 3

2. Los materiales (clave y descripción) que han sido proporcionados por el proveedor "Acme tools".














```
SELECT m.clave, m.descripcion
FROM materiales m, entregan e, proveedores p
WHERE p.rfc=e.rfc
AND m.clave=e.clave
AND p.razonsocial='Acme tools';
```

clave	descripcion
-------	-------------

Renglones: 0

3. El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales.

```
SELECT p.rfc
FROM proveedores p, entregan e
WHERE p.rfc = e.rfc
AND e.fecha BETWEEN '2000-01-01' AND '2001-01-01'
GROUP BY p.rfc
HAVING SUM(e.cantidad) >= 300;
```

	rfc
<input type="checkbox"/>  Editar  Copiar  Borrar	BBBBB800101
<input type="checkbox"/>  Editar  Copiar  Borrar	CCCCC800101
<input type="checkbox"/>  Editar  Copiar  Borrar	FFFFF800101
<input type="checkbox"/>  Editar  Copiar  Borrar	GGGGG800101

Renglones: 4

4. El Total entregado por cada material en el año 2000.

```
SELECT m.descripcion, SUM(e.cantidad)
FROM entregan e, materiales m
WHERE e.clave=m.clave
AND e.fecha BETWEEN '2000-01-01' AND '2001-01-01'
GROUP BY m.clave, m.descripcion
```


<div><div><div><div><div></div><div></div></div><div></div><div></div></div></div><div></div></div> <div>descripcion</div> <div>SUM(e.cantidad)</div>								
<input type="checkbox"/>		Editar		Copiar		Borrar	Varilla 3/17	8
<input type="checkbox"/>		Editar		Copiar		Borrar	Varilla 4/34	623
<input type="checkbox"/>		Editar		Copiar		Borrar	Block	466
<input type="checkbox"/>		Editar		Copiar		Borrar	Sillar gris	625
<input type="checkbox"/>		Editar		Copiar		Borrar	Cantera blanca	583
<input type="checkbox"/>		Editar		Copiar		Borrar	Recubrimiento P1028	43
<input type="checkbox"/>		Editar		Copiar		Borrar	Tubería 3.6	72
<input type="checkbox"/>		Editar		Copiar		Borrar	Pintura C1010	265
<input type="checkbox"/>		Editar		Copiar		Borrar	Pintura B1021	107
<input type="checkbox"/>		Editar		Copiar		Borrar	Pintura B1021	601
<input type="checkbox"/>		Editar		Copiar		Borrar	Pintura B1022	13

Renglones: 11

5. La Clave del material más vendido durante el 2001. (se recomienda usar una vista intermedia para su solución)

CREATE VIEW matMasVen

AS SELECT m.clave, m.descripcion, SUM(e.cantidad) AS TotalEnt

FROM materiales m, entregan e

WHERE m.clave=e.clave

AND e.fecha BETWEEN '2001-01-01' AND '2002-01-01'

GROUP BY m.clave, m.descripcion;

SELECT clave FROM matMasVen LIMIT 1;

clave

1000

Renglones: 1

6. Productos que contienen el patrón 'ub' en su nombre.

SELECT *

FROM materiales


























WHERE descripcion LIKE ('%ub%');

7. Denominación y suma del total a pagar para todos los proyectos.

SELECT pr.denominacion, SUM(e.cantidad*m.precio*(1 + m.porcentajeimpuesto / 100)) AS TotalPagar

FROM materiales m, proyectos pr, entregan e

WHERE m.clave=e.clave AND pr.numero=e.numero
GROUP BY denominacion

		denominacion	TotalPagar
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Ampliación de la carretera a la huasteca	578970.509
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Aztecon	150200.21899999998
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		CIT Campeche	161603.082
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		CIT Yucatan	871522.2999999998
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Construcción de Hospital Infantil	147860.80299999999
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Construcción de plaza Magnolias	125890.62340000001
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Disco Atlantic	162221.662
<input type="checkbox"/>	 Editar	 Copiar	 Borrar
		Educando en Coahuila	634846.174

Renglones: 20

8. **Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Solo usando vistas).**

CREATE VIEW Televisa

AS SELECT p.rfc, p.razonsocial

FROM proveedores p, proyectos pr, entregan e

WHERE p.rfc=e.rfc AND e.numero=pr.numero

AND pr.denominacion='Televisa en accion';

```
CREATE VIEW ProvEd
AS SELECT p.rfc, p.razonsocial
FROM proveedores p, proyectos pr, entregan e
WHERE p.rfc=e.rfc AND e.numero=pr.numero
AND pr.denominacion='Educando en Coahuila';
SELECT * FROM Televisa
EXCEPT
SELECT * FROM ProvEd;
```

rfc	razonsocial
CCCC800101	La Ferre
DDDD800101	Cecoferre

Renglones: 2

9. **Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Sin usar vistas, utiliza not in, in o exists).**

```
SELECT DISTINCT p.razonsocial, p.rfc
FROM proveedores p, entregan e, proyectos pr
WHERE p.rfc = e.rfc AND e.numero = pr.numero
AND pr.denominacion = 'Televisa en accion'
AND p.rfc NOT IN (
    SELECT p.rfc
    FROM proveedores p, entregan e, proyectos pr
    WHERE p.rfc = e.rfc
    AND e.numero = pr.numero
    AND pr.denominacion = 'Educando en Coahuila');
```

	razonsocial	rfc
<input type="checkbox"/>  Editar  Copiar  Borrar	La Ferre	CCCC800101
<input type="checkbox"/>  Editar  Copiar  Borrar	Cecoferre	DDDD800101

Renglones: 2

10. **Costo de los materiales y los Materiales que son entregados al proyecto Televisa en acción cuyos proveedores también suministran materiales al proyecto Educando en Coahuila.**

```
SELECT m.descripcion, SUM(e.cantidad * m.precio * (1 +
m.porcentajeimpuesto / 100)) AS CostoMat
FROM materiales m, proyectos pr, entregan e
WHERE m.clave = e.clave AND pr.numero= e.numero
AND pr.denominacion = 'Televisa en accion'
```

```
AND e.rfc IN (
    SELECT e.rfc
    FROM entregan e, proyectos pr
    WHERE e.numero = pr.numero
    AND pr.denominacion = 'Educando en Coahuila'
)
GROUP BY m.descripcion;
```

				descripcion	CostoMat			
<input type="checkbox"/>		Editar		Copiar		Borrar	Ladrillos rojos	4392.88
<input type="checkbox"/>		Editar		Copiar		Borrar	Tepetate	3731.1328000000003

Renglones: 2

11. Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos.

```
SELECT m.descripcion, COUNT(e.cantidad) AS CantEnt,
SUM(e.cantidad * m.precio * (1 + m.porcentajeimpuesto / 100)) AS
Costo
FROM materiales m, entregan e
WHERE m.clave = e.clave
GROUP BY m.descripcion;
```

descripcion	CantEnt	Costo
Arena	2	106169.28
Block	2	35718.9
Cantera amarilla	2	54844.006
Cantera blanca	2	164057.12
Cantera gris	2	1127663.13
Cantera rosa	2	589895.264
Cemento	2	258813.96
Grava	2	78105
Gravilla	2	100664.38799999999

Renglones: 42